Appl. No. 09/538,132
Amdt. sent July 2, 2004
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group

PATENT

REMARKS/ARGUMENTS

Claims 1 - 54 are pending.

Claim 11 was rejected under 35 U.S.C. § 102(e) for allegedly being anticipated by Riddle et al., U.S. Patent No. 6,457,051.

The following rejections were raised under 35 U.S.C. § 103 for alleged obviousness in view of the associated references:

- claims 1, 2, 4, and 12, Riddle and Tang, U.S. Patent No. 5,378,126;
- claims 3 and 18 - 22, Riddle, Tang, and Moreno, U.S. Patent No. 5,951,674;
- claims 5 - 7 and 9, Riddle, Tang, and Del Monte, U.S. Patent No. 5,704,060;
- claims 8, 16, and 17, Riddle, Tang, Del Monte, and Eager et al., U.S. Patent No. 5,960,200
- claim 10, Riddle, Tang, Del Monte, and Moreno;
- claims 13 - 15, Riddle and Del Monte; and
- claims 30 and 37 - 41, Riddle, Tang, Del Monte, and Boucher, U.S. Patent No. 6,226,680.

Claims 23 - 29, 31 - 36, and 42 - 54 were alleged not to add any new limitations to above claims 1 - 22 and claims 37 - 41, and therefore were rejected for similar reasons.

The foregoing rejections were made final.

Counsel for Applicant is gracious for the Examiner's time and attention during a telephonic discussion on June 30, 2004. Independent claims 1, 11, 23, 31, 40, 47, and 51, and dependent claim 30 have been amended per the discussion. No new matter is believed to have been introduced by the claim amendments. The primary reference to Riddle et al. is believed to be distinguished, and the pending claims are believed to be allowable over the prior art.

Appl. No. 09/538,132
Amdt. sent July 2, 2004
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group

PATENT

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 650-326-2400.

Respectfully submitted,

George B. F. Yee
Reg. No. 37,478

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
GBFY:cmm
60189768 v1

Page 1 of 6   telephonic interview: June 30, 2004

TO: Examiner Hussein A. El Chanti
FAX: (703) 746 - 9679

Re: 09/538,132 (27-1, PMC 981041)
       09/557,736 (26-1, PMC 990324)
       Office actions, both mailed March 30, 2004
       Telephonic interview to be conducted June 30, 2004 (5 PM EST)

Dear Examiner El Chanti:

Following is an informal overview that you asked for to facilitate your preparations for
Wednesday's interview.

We would like to discuss with you to understand in what way the present claims are deemed too
broad, and to discuss possible language that might be added to limit the scope of the claims.

Following are discussion points we would like to pursue. These points address portions of the
discussion in part 11 (on page 12) in each of the most recent Office actions.

For convenience, the pending independent claims for each application are provided.

DISCUSSION POINTS

A. Riddle (col. 13 lines 57 - col. 14 lines 9, and figure 4A step 402) describes steps to follow
the classification tree. The classification tree does not constitute the "lexical parsing
rules" of the present invention. Also, Riddle (col 14 lines 50 - col 15 lines 20) does not
mention anything about regular expressions.

B. Riddle (col 14 lines 50 - col 15 lines 20) does not describe a "lexical token" that is a
result of DFA parsing. Instead, it describes in Fig. 4A at step 408, a function of
producing "a list of characteristics of traffic". The LIST (containing multiple items)
mentioned by Riddle is different from the result of the lexical parser that generates only a
SINGLE lexical TOKEN.

C. The cited portion of Riddle at col 14 lines 50 - col 15 lines 20 does not suggest applying
the techniques of a grammar or a parser for classifying IP packets. Generally, Riddle
does not disclose a lexical technique, and so the remaining cited references used in
various combinations with Riddle cannot be combined to render obvious the present
invention.

George Yee
(650) 324 - 6352

INDEPENDENT CLAIMS FOR 09/538,132 (27-1, PMC 981041)

1.   A method for classifying received network data comprising: scanning incoming network data using lexical token scanning wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said lexical scanning resulting in the identification of a data packet as belonging to one of a plurality of classes.

11.   In a network data switching device, a method for classifying data packets comprising steps of:

providing one or more regular expressions, each having an associated class identifier;

receiving plural data packets, each having a length not necessarily equal to one another; and

for each data packet, determining a matching one of said regular expressions that matches said data packet, wherein said each data packet is classified according to the class identifier associated with said matching regular expression.

23.   In a data packet receiving and forwarding device, a method for classifying received data packets comprising a stream of data, said method comprising steps of:

receiving a description of classification rules in a classification language;

compiling said classification language to produce a deterministic finite automaton (DFA) comprising plural states;

configuring a programmable hardware packet classifier with said DFA; and

scanning said data stream with said hardware packet classifier to classify said received data packets.

31.   A network data packet classifier comprising:

an input port for receiving network data packets comprising a stream of data;

a memory assemblage configured with data representing a deterministic finite automaton (DFA), said DFA representing plural regular expressions; and

decompression logic operatively coupled to said memory assemblage and configured to scan said stream of data with said DFA to find a matching one of said regular expressions,

said regular expressions having corresponding class identifiers,

wherein each of said network data packets is associated with the class identifier of said regular expression that matches it.

40.   A network data packet classifier comprising:

an input configured to provide a data packet comprising a stream of data;

a first system of memory configured with data representing a deterministic finite automaton (DFA), said DFA comprising plural states including an initial state and plural terminating states;

a system of logic circuits operatively coupled to said first system of memory and to said input, and configured to lexically scan said data stream with said DFA to produce a reached terminating state; and

a second system of memory configured with data representing a class index corresponding to each of said terminating states and configured to output a class index in response to the production of said reached terminating state.

47.   A network packet classifier comprising:

means for receiving an incoming network packet;

means for classifying said network packet by matching the pattern of its constituent data against plural regular expressions, each regular expression having a corresponding class identifier; and

means for outputting a class identifier of the regular expression which matches said network packet.

51.   A network packet classifier comprising:

a dual-ported memory component;

first classification logic operatively coupled to a first port of said dual-ported memory component and having a first input for receiving a data stream; and

Page 4 of 6 telephonic interview: June 30, 2004

second classification logic operatively coupled to a second port of said dual-ported memory component and having a second input for receiving a data stream,

said memory component configured to contain a deterministic finite automaton (DFA) comprising plural states,

said DFA representing plural regular expressions for matching data packets,

said first and second classification logic each configured to scan its associated data stream using said DFA to identify data packets contained therein and to classify identified data packets.

telephonic interview: June 30, 2004

INDEPENDENT CLAIMS FOR 09/557,736 (26-1, PMC 990324)

1. A method for identifying protocol encapsulation in received network data comprising providing a grammar and parsing incoming network data using said grammar, said network data being organized into data packets.

6. In a data packet network switching device, a method for processing data packets comprising:

providing a grammar;

receiving plural data packets, each having a length not necessarily equal to one another; and

for each data packet, lexically scanning said data packet to produce plural lexical tokens, parsing said lexical tokens using said grammar to produce one or more identified protocols, and processing said data packet based on said identified protocols.

11. In a data packet receiving and forwarding device, a method for processing data packets comprising a stream of data, said method comprising:

receiving a description of grammar rules in a grammar packet classification language;

compiling said grammar packet classification language to produce a grammar graph;

configuring a programmable grammatical packet classifier with said grammar graph;

parsing said data stream with said grammatical packet classifier to identify a protocol structure in a received data packet; and

processing said received data packet in accordance with said protocol structure.

18. A network data packet classifier comprising:

an input port for receiving network data packets comprising a stream of data;

a memory assemblage configured with data representing a deterministic finite automaton (DFA), said DFA representing a grammar graph and plural regular expressions; and

Page 6 of 6   telephonic interview: June 30, 2004

decompression logic operatively coupled to said memory assemblage and configured to scan said stream of data with said DFA to find a matching one of said regular expressions thereby producing plural lexical tokens,

said decompression logic further configured to parse said lexical tokens with said DFA to identify a protocol structure in a received network data packet,

wherein processing of said network data packet depends on said protocol structure.

27.   A network packet classifier comprising:

means for receiving an incoming network packet; and

means for identifying protocol structure in said network packet including means for scanning to match patterns in its constituent data against plural regular expressions to produce lexical tokens and means for parsing through said lexical tokens using a grammar.